On Improving the Convergence of Radau IIA Methods Applied to Index 2 DAEs

Anne Aubry, Philippe Chartier

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

On improving the convergence of Radau IIA methods applied to index 2 DAEs

A. Aubry , P. Chartier

N° 2744

Décembre 1995

PROGRAMME 6

ISSN 0249-6399

INRIA
RENNES

On improving the convergence of Radau IIA methods applied to index 2 DAEs

A. Aubry , P. Chartier*

Programme 6 — Calcul scientifique, modélisation et logiciel numérique
Projet Aladin

Rapport de recherche n° 2744 — Décembre 1995 — 24 pages

Abstract: This paper presents a new simple technique to improve the order behaviour of Runge-Kutta methods when applied to index 2 DAEs. It is then shown how this can be incorporated into a more efficient version of the code RADAU5 developed by E. Hairer and G. Wanner.

Key-words: differential-algebraic systems of index 2, Runge-Kutta methods, Radau IIA methods, rooted trees, simplifying assumptions, composition.

(Résumé : tsvp)

*INRIA-Rennes, Campus de Beaulieu, 35042 Rennes Cedex, FRANCE

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : (33) 99 84 71 00 – Télécopie : (33) 99 84 71 71

Augmenter l'ordre de convergence des composantes algébriques pour des méthodes de Radau IIA appliquées à des EDA d'incice deux

Résumé : Cet article présente une nouvelle technique simple pour améliorer la convergence des composantes algebriques lorsqu'une méthode de Radau de type IIA est appliquée à un système différentiel algébrique d'indice deux. Pour la méthode de Radau IIA d'ordre 5, sa mise en œuvre recquiert de simples modifications du code Radau5 développé par E. Hairer et G. Wanner. Nous les explicitons et présentons les améliorations obtenues sur différents problèmes.

Mots-clé : systèmes algébro-différentiels d'indice 2, méthodes de Runge-Kutta, méthodes de Radau IIA, arbres, conditions simplificatrices, composition.

1 Introduction

In recent years, differential algebraic equations (DAEs) have been studied by various authors (see [HW91], [HLR89], [BCP91]), and their importance acknowledged by the development of specific solvers such as DASSL from [Pet86] or RADAU5 from [HW91]. An especially important class of DAEs arising in practice are semi-explicit systems of the form

$$(S)\ \begin{cases} y'(t) &= f(y(t),z(t)), \\ 0 &= g(y(t)), \end{cases} \quad t \in [t_b, t_e],$$

where $g_y f_z$ is assumed to be of bounded inverse in a neighbourhood of the solution of (S). Here, we are interested in obtaining a numerical approximation to (S) accurate for both the differential and the algebraic components. Although some of the ideas presented in this paper also apply to more general Runge-Kutta methods, we will focus on Radau IIA methods, given that they were used to build the code RADAU5. Their construction as well as some of their properties are briefly recalled in Section 1.1.

When applying a s-stage Radau IIA method to (S), the orders of convergence are respectively $2s-1$ for the y-component and s for the z-component (see [HLR89]). In some situations, where getting an accurate value of z may be important (in mechanics for instance), one is led to use a different approach. Generally speaking, the order reduction phenomenon may be overcome by the following techniques:

(i) a first possibility consists in applying the Radau IIA method to the index one formulation $(\overline{S})$ of (S),

$$(\overline{S})\ \begin{cases} y' &= f(y,z), \\ 0 &= g_y(y)f(y,z). \end{cases}$$

Since Radau IIA methods applied to index one DAEs exhibit full order of convergence for y and z (see Theorem 3-1 [HLR89]), the order of convergence is now $2s-1$ also for z. However, solving $(\overline{S})$ can be considerably more costly: as a matter of fact, this requires to evaluate the Jacobian of the function $\overline{F}(y,z) = (f(y,z), g_y(y)f(y,z))$ at each step (or whenever the convergence rate of the Newton iteration gets too small), instead of the function $F(y,z) = (f(y,z), g(y))$. Another drawback is that the numerical solution is not forced any longer to lie on the constraint manifold $g(y) = 0$.

(ii) a second idea consists in computing the z-component by solving the additional equation $g_y(y)f(y,z)$, i.e. in projecting the numerical solution on the so-called "hidden constraint". The corresponding numerical scheme now reads

$$(S_n)\ \begin{cases} 0 &= g(y_{n,i}), \\ y_{n,i} &= y_n + h_n \sum_{j=1}^{s} a_{ij} f(y_{n,j}, z_{n,j}), \\ y_{n+1} &= y_n + h_n \sum_{i=1}^{s} b_i f(y_{n,i}, z_{n,i}), \\ 0 &= g_y(y_{n+1}) f(y_{n+1}, z_{n+1}). \end{cases} \quad i = 1, \cdots, s,$$

The order of convergence for the y-component is still $2s-1$ and can be shown to be now $2s-1$ also for the z-component by using the Implicit Function Theorem. However, this technique is once again computationally more demanding than the original one: solving the new implicit part of (S_n) requires an accurate evaluation of $g_y(y)$ at each step, and not only, as previously mentioned, whenever the convergence rate of the Newton iteration becomes too small. It can be nevertheless noted that in a parallel environment, this additional cost would be shadowed by the use of a second processor.

In this paper, we present a third approach which does not require an analytical form of g_y and whose computational cost is basically equal to what it is for the standard formulation. It is based on the

observation that the errors in the z-component are essentially of a local nature, at least up to the order of convergence of the y-component. As a consequence, making z more accurate is a matter of recovering the significant terms that appear in the so-called "B-series of the error". This is made possible by considering the composition of the basic method with itself over several steps. It can be noted that similar ideas were used by R.P.K. Chan to deal with the order reduction of Gauss methods when applied to certain stiff problems (see [BC93]).

The new order conditions will be determined in Section 2. While they can be derived in a straightforward manner from the work of E. Hairer, C. Lubich and M. Roche ([HLR89]), they are still relatively unknown since they are not satisfied by most classical Runge-Kutta methods (see also [BC93] or [CC95]). It will then be shown that some of those conditions are actually redundant and may be omitted. This is a crucial aspect of the method, since it allows the construction of formulas with a manageable level of complexity.

In Section 3, the implied modifications to the code RADAU5 are listed. Finally, numerical results are presented that illustrate the advantages of this new technique.

1.1 Basic properties of Radau IIA methods

Radau IIA methods can be defined by the $s+1$ quadrature formulas

$$\begin{array}{lcl} \int_0^{c_i} \phi(t)dt & \approx & \sum_{j=1}^{s} a_{ij}\phi(c_i), \quad i = 1, \cdots, s, \\ \int_0^{1} \phi(t)dt & \approx & \sum_{i=1}^{s} b_i\phi(c_i). \end{array}$$

A particular method $\mathcal{R}$ will be characterized by the triple (A, b, c) where $(a_{ij})_{i,j=1,\cdots,s}$ is a $s \times s$ matrix, $b = (b_1, \cdots, b_s)^T$ a s-dimensional vector and $c = (c_1, \cdots, c_s)^T$ a s-dimensional vector. In the sequel, we will furthermore use the notations $e = (1, \cdots, 1)^T \in I\!R^s$ and $c^k = (c_1^k, \cdots, c_s^k)^T$ for all integers k.

1.1.1 Construction of Radau IIA methods

Their coefficients are uniquely determined by the following conditions:

1. $c_1, \cdots, c_s$ are the ordered zeros of the Radau right polynomial

$$M(x) = \frac{d^{s-1}}{dx^{s-1}}\left(x^{s-1}(x-1)^s\right).$$

2. $b_1, \cdots, b_s$ satisfy $B(s)$:

$$\forall k \in \{0, \cdots, s-1\},\ b^T c^k = \frac{1}{(k+1)}.$$

3. The coefficients a_{ij} of the matrix A satisfy $C(s)$:

$$\forall k \in \{0, \cdots, s-1\},\ Ac^k = \frac{1}{(k+1)}\, c^{k+1}.$$

   As the c_i are all distinct, A is non-singular.

1.1.2 Some useful properties

We will refer here to the additional simplifying assumptions $D(\xi)$ introduced, as $B(p)$ and $C(\eta)$ of previous subsection, by J.C. Butcher (see [HW91] page 75).

1. Due to the conditions $C(s)$, $B(s)$ and $c_s = 1$, Radau IIA methods are stiffly accurate (i.e. $b_i = a_{si}$, $i = 1, \cdots, s$). (The vectors b and $(a_{s1}, \cdots, a_{ss})^T$ are solution of the same Cramer system).

2. They are collocation methods (Theorem 7-8 [HNW91], page 212).

3. $B(2s-1)$ is satisfied (Lemma 5-15 [HW91], page 93).

4. $D(s-1)$ is satisfied (Lemma 5-4 [HW91], page 78).

Optimal convergence results have been obtained for those methods by J.C. Butcher on the one hand (Theorem 5-3 of [HW91]) and E. Hairer, C. Lubich and M. Roche on the other hand (Theorems 3-1, 4-4 and 4-6 of [HLR89]). They are collected in Table 1.

	index 0 (ODE)	index 1	index 2
y	h^{2s-1}	h^{2s-1}	h^{2s-1}
z	-	h^{2s-1}	h^s

Table 1: Optimal global error estimates for the s-stage method Radau IIA

1.2 Increasing the order of convergence of the z-component

When applying a s-stage Radau IIA method $\mathcal{R} = (A, b, c)$ to the system (S), we obtain

$$\begin{cases} 0 &= g(y_{n,i}), & (1) \\ y_{n,i} &= y_n + h_n \sum_{j=1}^s a_{ij} f(y_{n,j}, z_{n,j}), \quad i = 1, \cdots, s, & (2) \\ y_{n+1} &= y_n + h_n \sum_{i=1}^s b_i f(y_{n,i}, z_{n,i}), & (3) \\ z_{n+1} &= \sum_{i,j=1}^s b_i \omega_{ij} z_{n,j}, & (4) \end{cases}$$

where ω_{ij} are the coefficients of the matrix A^{-1}. In (4), z_n vanishes because $R(\infty) = 0$. Let us now replace the vector $(b^T A^{-1})^T$ by an adjustable vector $w = (w_1, \cdots, w_s)^T$ in (4). By doing so, we define a new method $\mathcal{R}_w$. It is easily seen that the order of convergence of the y-component remains unchanged. As for the z-component, the lack of accumulation makes the errors purely local. The convergence behaviour of the z-component is thus entirely determined by the following order conditions from Theorem 8-6 and 8-8 of [HW91].

Proposition 1 *Let δ_y and δ_z be the local errors respectively for the y and z components of a Runge-Kutta method. Then we have,*

$$\begin{aligned} \delta y = O(h^{p+1}) \quad & \textit{iff} \quad \forall t \in DAT2_y, \rho(t) \leq p, \gamma(t) b^T \Phi(t) = 1, \\ \delta z = O(h^{q+1}) \quad & \textit{iff} \quad \forall u \in DAT2_z, \rho(u) \leq q, \gamma(u) w^T \Phi(u) = 1, \end{aligned}$$

where $DAT2_y$, $DAT2_z$ are sets of trees, Φ a vectorial function and γ, ρ scalar ones associated with the trees [1].

Nevertheless, w has not enough components to allow for an order of convergence greater than s ($w = (b^T A^{-1})^T$ is the optimal vector). Hence, to get sufficient freedom, we need to consider the composition $\mathcal{R}_\sigma$ of $\mathcal{R}$ over σ steps. As variable steps h_i, $i = 1, \cdots, \sigma$ are considered, we also have to introduce the ratios $r_i = h_i / \sum_{i=1}^\sigma h_i$, $i = 1, \cdots, \sigma$. $\mathcal{R}_\sigma$ is characterized by the triple $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ where

- $\mathcal{A}$ is the blockmatrix $(A_{ij})_{i,j=1,\cdots,\sigma}$ with $s \times s$ blocks of the form

[1] See [HW91] for a definition of these notions.

$$\begin{aligned} A_{ii} &= r_i A, \\ A_{ij} &= r_j e b^T, \quad \text{if } i > j, \\ A_{ij} &= 0, \quad \text{if } j > i. \end{aligned}$$

- $\mathcal{B}$ is the blockvector $(B_i)_{i=1,\cdots,\sigma}$ with blocks of the form $B_i = r_i b$.
- $\mathcal{C}$ is the blockvector $(C_i)_{i=1,\cdots,\sigma}$ with blocks of the form $C_i = r_i c + \left(\sum_{k=1}^{i-1} r_k\right) e$.

Replacing $\left(\mathcal{B}^T \mathcal{A}^{-1}\right)^T$ by an adjustable vector w offers $\sigma \times s$ degrees of freedom, i.e hopefully enough for $\sigma > 1$ to increase the order of convergence of the z-component. It is our aim now to show how to construct w and how to implement the new method $\mathcal{R}_{\sigma,w}$.

2 Construction of the vector w

In this section, the effective construction of w is described. It should be emphasized that its components depend on $r_1, \cdots, r_s$, thus forcing one evaluation per step. However, these additional computations become negligible as soon as the dimension of the system (S) is large enough.

2.1 Order conditions

The conditions for order k are enumerated below together with the associated trees.

- $k = s$, $s \geq 1$: (C_s) is required with

$$(C_s) \left\{ \begin{array}{lllll} & w^T \mathcal{E} & = 1, & & (0) \\ \forall k \in \{1, \cdots, s-1\} & w^T \mathcal{C}^k & = 1, & [[\overbrace{\tau, \cdots, \tau}^{k}]_y]_z. & (k) \end{array} \right.$$

- $k = s+1$, $s \geq 2$: (C_s) and (C_{s+1}) are required with

$$(C_{s+1}) \left\{ \begin{array}{llll} w^T \mathcal{C}^s & = 1, & [[\overbrace{\tau, \cdots, \tau}^{s}]_y]_z, & (s) \\ w^T \mathcal{A}^{-1} \mathcal{C}^{s+1} & = s+1, & [\overbrace{\tau, \cdots, \tau}^{s+1}]_z. & (s+1) \end{array} \right.$$

  Let U_s be $\mathcal{A}\mathcal{C}^s - \frac{1}{s+1}\mathcal{C}^{s+1}$. If (s) is satisfied, then (C_{s+1}) is equivalent to

$$w^T \mathcal{A}^{-1} U_s = 0.$$

- $k = s+2$, $s \geq 3$: (C_s), (C_{s+1}) and (C_{s+2}) are required with [2]

$$(C_{s+2}) \left\{ \begin{array}{llll} w^T \mathcal{C}^{s+1} & = 1, & [[\overbrace{\tau, \cdots, \tau}^{s+1}]_y]_z, & (s+2) \\ w^T \mathcal{A}^{-1} \mathcal{C}^{s+2} & = s+2, & [\overbrace{\tau, \cdots, \tau}^{s+2}]_z, & (s+3) \\ w^T \mathcal{A}\mathcal{C}^s & = \frac{1}{s+1}, & [[[\overbrace{\tau, \cdots, \tau}^{s}]_y]_y]_z, & (s+4) \\ w^T \mathcal{C}.\mathcal{A}^{-1} \mathcal{C}^{s+1} & = s+1, & [[\tau, [\overbrace{\tau, \cdots, \tau}^{s+1}]_z]_y]_z, & (s+5) \\ w^T \mathcal{A}^{-1}(\mathcal{C}.\mathcal{A}\mathcal{C}^s) & = \frac{s+2}{s+1}, & [\tau, [\overbrace{\tau, \cdots, \tau}^{s}]_y]_z. & (s+6) \end{array} \right.$$

  Let U_{s+1} be $\mathcal{A}\mathcal{C}^{s+1} - \frac{1}{s+2}\mathcal{C}^{s+2}$. If $(s+2)$ is satisfied, then (C_{s+2}) is equivalent to

[2] The dot stands for the componentwise product.

$$\begin{array}{llll} w^T\mathcal{A}^{-1}U_{s+1} = 0 \quad (s+3), & w^TU_s = 0 \quad (s+4), \\ w^T\mathcal{C}.\mathcal{A}^{-1}U_s = 0 \quad (s+5), & w^T\mathcal{A}^{-1}(\mathcal{C}.U_s) = 0 \quad (s+6). \end{array}$$

- $k = s+3$, $s \geq 4$: (C_s), (C_{s+1}), (C_{s+2}) and (C_{s+3}) are required with

$$(C_{s+3}) \left\{ \begin{array}{llll} w^T\mathcal{C}^{s+2} = 1, & [[\overbrace{\tau,\cdots,\tau}^{s+2}]_y]_z, & (s+7) \\ w^T\mathcal{A}^{-1}\mathcal{C}^{s+3} = s+3, & [\overbrace{\tau,\cdots,\tau}^{s+3}]_z, & (s+8) \\ w^T\mathcal{A}\mathcal{C}^{s+1} = \frac{1}{s+2}, & [[[\overbrace{\tau,\cdots,\tau}^{s+1}]_y]_y]_z, & (s+9) \\ w^T\mathcal{C}.\mathcal{A}^{-1}\mathcal{C}^{s+2} = s+2, & [[\tau,[\overbrace{\tau,\cdots,\tau}^{s+2}]_z]_y]_z, & (s+10) \\ w^T\mathcal{A}^{-1}(\mathcal{C}.\mathcal{A}\mathcal{C}^{s+1}) = \frac{s+3}{s+2}, & [\tau,[\overbrace{\tau,\cdots,\tau}^{s+1}]_y]_z, & (s+11) \\ w^T\mathcal{A}^{-1}(\mathcal{C}.\mathcal{A}^2\mathcal{C}^s) = \frac{s+3}{(s+2)(s+1)}, & [\tau,[[\overbrace{\tau,\cdots,\tau}^{s}]_y]_y]_z, & (s+12) \\ w^T\mathcal{A}^{-1}(\mathcal{C}.\mathcal{A}(\mathcal{C}.\mathcal{A}^{-1}\mathcal{C}^{s+1})) = \frac{(s+3)(s+1)}{s+2}, & [\tau,[\tau,[\overbrace{\tau,\cdots,\tau}^{s+1}]_z]_y]_z, & (s+13) \\ w^T\mathcal{A}^{-1}(\mathcal{C}^2.\mathcal{A}\mathcal{C}^s) = \frac{s+3}{s+1}, & [\tau,\tau,[\overbrace{\tau,\cdots,\tau}^{s}]_y]_z, & (s+14) \\ w^T\mathcal{A}^2\mathcal{C}^s = \frac{1}{(s+2)(s+1)}, & [[[[\overbrace{\tau,\cdots,\tau}^{s}]_y]_y]_y]_z, & (s+15) \\ w^T\mathcal{A}(\mathcal{C}.\mathcal{A}^{-1}\mathcal{C}^{s+1}) = \frac{s+1}{s+2}, & [[[\tau,[\overbrace{\tau,\cdots,\tau}^{s+1}]_z]_y]_y]_z, & (s+16) \\ w^T\mathcal{C}.\mathcal{A}\mathcal{C}^s = \frac{1}{s+1}, & [[\tau,[\overbrace{\tau,\cdots,\tau}^{s}]_y]_y]_z, & (s+17) \\ w^T\mathcal{C}.\mathcal{A}^{-1}(\mathcal{C}.\mathcal{A}\mathcal{C}^s) = \frac{s+2}{s+1}, & [[\tau,[\tau,[\overbrace{\tau,\cdots,\tau}^{s}]_y]_z]_y]_z, & (s+18) \\ w^T\mathcal{C}^2.\mathcal{A}^{-1}\mathcal{C}^{s+1} = s+1, & [[\tau,\tau,[\overbrace{\tau,\cdots,\tau}^{s+1}]_z]_y]_z. & (s+19) \end{array} \right.$$

Let U_{s+2} be $\mathcal{A}\mathcal{C}^{s+2} - \frac{1}{s+3}\mathcal{C}^{s+3}$. If $(s+7)$ is satisfied, then (C_{s+3}) is equivalent to

$$\begin{array}{llll} w^T\mathcal{A}^{-1}U_{s+2} = 0 \quad (s+8), & w^TU_{s+1} = 0 \quad (s+9), \\ w^T\mathcal{C}.\mathcal{A}^{-1}U_{s+1} = 0 \quad (s+10), & w^T\mathcal{A}^{-1}(\mathcal{C}.U_{s+1}) = 0 \quad (s+11), \\ w^T\mathcal{A}^{-1}(\mathcal{C}.\mathcal{A}U_s) = 0 \quad (s+12), & w^T\mathcal{A}^{-1}(\mathcal{C}.\mathcal{A}(\mathcal{C}.\mathcal{A}^{-1}U_s)) = 0 \quad (s+13), \\ w^T\mathcal{A}^{-1}(\mathcal{C}^2.U_s) = 0 \quad (s+14), & w^T\mathcal{A}U_s = 0 \quad (s+15), \\ w^T\mathcal{A}(\mathcal{C}.\mathcal{A}^{-1}U_s) = 0 \quad (s+16), & w^T\mathcal{C}.U_s = 0 \quad (s+17), \\ w^T\mathcal{C}.\mathcal{A}^{-1}(\mathcal{C}.U_s) = 0 \quad (s+18), & w^T\mathcal{C}^2.\mathcal{A}^{-1}U_s = 0 \quad (s+19). \end{array}$$

These conditions are obtained by Proposition 1 and by using simplifying assumptions (it is important to note that the composite method $\mathcal{R}_\sigma$ satisfies $B(2s-1)$, $C(s)$ and $D(s-1)$).

2.2 Preliminary calculus

In order to later simplify the equations for w, we now state some basic results.

Lemma 1 *Let F be $A^{-1}EA^{-1}$ with $E = eb^T$, then $F(I - EA^{-1}) = 0$.*

Proof: By definition, we have $F = A^{-1}eb^TA^{-1}$. As the method $\mathcal{R} = (A, b, c)$ is stiffly accurate, $F = ve_s^T$ where $v = A^{-1}e$ and $e_s = (0, \cdots, 0, 1)^T$. Hence, $F(I - EA^{-1}) = ve_s^T(I - eb^TA^{-1}) = ve_s^T - ve_s^Tee_s^T = ve_s^T - ve_s^T = 0$. □

Lemma 2 *Let $(W_{i,j})_{i,j=1,\cdots,\sigma}$ be the $s \times s$ blocks of the matrix $\mathcal{A}^{-1}$, then*

$$\begin{array}{lcll} W_{i,i} & = & \frac{1}{r_i}A^{-1}, & \\ W_{i,i-1} & = & -\frac{1}{r_i}F, & \\ W_{i,j} & = & 0, & \textit{if } j \neq i \textit{ and } j \neq i-1. \end{array}$$

Proof: By definition, $\mathcal{A}\mathcal{A}^{-1} = \mathcal{I}$, i.e.

$$\forall i,j \in \{1,\cdots,\sigma\},\ r_i A W_{i,j} + \sum_{k=1}^{i-1} r_k E W_{k,j} = \delta_{ij} I.$$

Lemma 2 is then easily proved by induction on i. □

Lemma 3 *Let $U_n = \left(U_{1,n}^T,\cdots,U_{\sigma,n}^T\right)^T$ be the $\sigma \times s$ dimensional vector $\mathcal{A}\mathcal{C}^n - \frac{1}{n+1}\mathcal{C}^{n+1}$ and u_n be the s dimensional vector $Ac^n - \frac{1}{n+1}c^{n+1}$ with $n \in I\!N$. Then, for all integer $n \leq 2s-2$, U_n is given by*

$$\forall i \in \{1,\cdots,\sigma\},\ U_{i,n} = \sum_{k=s}^{n} \left(\begin{array}{c} n \\ k \end{array} \right) r_i^{k+1} s_i^{n-k} u_k.$$

Proof: Let n be less than or equal to $2s-2$. For all $i \in \{1,\cdots,\sigma\}$, we have $U_{i,n} = T_{i,n} + S_{i,n}$, where

$$\begin{array}{lcl} T_{i,n} & = & \sum_{k=1}^{i-1} r_k E \left(r_k c + s_k e\right)^n, \\ S_{i,n} & = & r_i A \left(r_i c + s_i e\right)^n - \frac{1}{n+1}\left(r_i c + s_i e\right)^{n+1}. \end{array}$$

$S_{i,n}$ can be expanded as follows

$$\begin{array}{rcl} S_{i,n} & = & \sum_{k=0}^{n} \left(\begin{array}{c} n \\ k \end{array} \right) r_i^{k+1} s_i^{n-k} A c^k - \frac{1}{n+1}\sum_{k=0}^{n+1} \left(\begin{array}{c} n+1 \\ k \end{array} \right) r_i^k s_i^{n+1-k} c^k \\ & = & \sum_{k=0}^{n} \left(\begin{array}{c} n \\ k \end{array} \right) r_i^{k+1} s_i^{n-k} u_k - \frac{1}{n+1} s_i^{n+1} e. \end{array}$$

Owing to $C(s)$, we have $u_k = 0$ for all integer $k < s$, hence

$$S_{i,n} = \sum_{k=s}^{n} \left(\begin{array}{c} n \\ k \end{array} \right) r_i^{k+1} s_i^{n-k} u_k - \frac{1}{n+1} s_i^{n+1} e.$$

Similarly, $T_{i,n}$ can be expanded as

$$T_{i,n} = \sum_{k=1}^{i-1} r_k E \left(\sum_{l=0}^{n} \left(\begin{array}{c} n \\ l \end{array} \right) r_k^l s_k^{n-l} c^l \right).$$

Now, $B(2s-1)$ implies that

$$\forall l \in \{0,\cdots,2s-2\},\ E c^l = e b^T c^l = \frac{1}{l+1} e,$$

and as $n \leq 2s-2$, we obtain

$$\begin{array}{rcl} T_{i,n} & = & \frac{1}{n+1}\left(\sum_{k=1}^{i-1}\sum_{l=0}^{n} \left(\begin{array}{c} n+1 \\ l+1 \end{array} \right) r_k^{l+1} s_k^{n-l} \right) e \\ & = & \frac{1}{n+1}\left(\sum_{k=1}^{i-1} \left(r_k + s_k\right)^{n+1} - s_k^{n+1} \right) e \\ & = & \frac{1}{n+1}\left(\sum_{k=1}^{i-1} s_{k+1}^{n+1} - s_k^{n+1} \right) e \\ & = & \frac{1}{n+1} s_i^{n+1} e. \end{array}$$

□

Lemma 4 *For all n less than $2s-2$, $Fu_n = 0$.*

Proof: This follows at once from $B(2s-1)$. □

Lemma 5 *Let $\left(X_{1,n}^T, \cdots, X_{\sigma,n}^T\right)^T$ be the vector $\mathcal{A}^{-1}U_n$, then for all integer $n \leq 2s-2$, $\mathcal{A}^{-1}U_n$ is given by*

$$\forall i \in \{1, \cdots, \sigma\},\ X_{i,n} = \sum_{k=s}^{n} \begin{pmatrix} n \\ k \end{pmatrix} r_i^k s_i^{n-k} A^{-1} u_k.$$

Proof: By definition, $X_{i,n} = \sum_{j=1}^{\sigma} W_{i,j} U_{j,n}$. From Lemma 2, $X_{i,n} = -\frac{1}{r_i} F U_{i-1,n} + \frac{1}{r_i} A^{-1} U_{i,n}$. Using Lemma 3, we obtain

$$X_{i,n} = -\frac{1}{r_i} \sum_{k=s}^{n} \begin{pmatrix} n \\ k \end{pmatrix} r_{i-1}^{k+1} s_{i-1}^{n-k} F u_k + \sum_{k=s}^{n} \begin{pmatrix} n \\ k \end{pmatrix} r_i^k s_i^{n-k} A^{-1} u_k,$$

and we use Lemma 4 to complete the proof. □

Lemma 6 *For all n less than $2s-3$, $F(c.u_n) = 0$.*

Proof: This follows straightforwardly from the order conditions for the trees $[[\overbrace{\tau, \cdots, \tau}^{n+2}]_z]_y$ and $[[\tau, [\overbrace{\tau, \cdots, \tau}^{n}]_y]_z]_y$. □

Lemma 7

1. *Let $\left(Y_{1,n}^T, \cdots, Y_{\sigma,n}^T\right)^T$ be the vector $\mathcal{C}.\mathcal{A}^{-1}U_n$;*

$$\text{if } n \leq 2s-2 \text{ then } Y_{i,n} = \sum_{k=s}^{n} \begin{pmatrix} n \\ k \end{pmatrix} \left(r_i^{k+1} s_i^{n-k} c.A^{-1} u_k + r_i^k s_i^{n+1-k} A^{-1} u_k \right).$$

2. *Let $\left(Z_{1,n}^T, \cdots, Z_{\sigma,n}^T\right)^T$ be the vector $\mathcal{A}^{-1}(\mathcal{C}.U_n)$;*

$$\text{if } n \leq 2s-3 \text{ then } Z_{i,n} = \sum_{k=s}^{n} \begin{pmatrix} n \\ k \end{pmatrix} \left(r_i^{k+1} s_i^{n-k} A^{-1}(c.u_k) + r_i^k s_i^{n+1-k} A^{-1} u_k \right).$$

Proof: By definition, $Y_{i,n} = (r_i c + s_i e)\, X_{i,n}$ and the first part of the result is obtained by applying Lemma 5. Now, let $\left(T_{1,n}^T, \cdots, T_{\sigma,n}^T\right)^T$ denote the vector $\mathcal{C}.U_n$. From Lemma 3, we can write $T_{i,n}$ as

$$T_{i,n} = \sum_{k=s}^{n} \begin{pmatrix} n \\ k \end{pmatrix} \left(r_i^{k+2} s_i^{n-k} c.u_k + r_i^{k+1} s_i^{n+1-k} u_k \right).$$

We have furthermore $Z_{i,n} = \sum_{j=1}^{\sigma} W_{i,j} T_{j,n}$, so that Lemma 2 leads to

$$\begin{aligned} Z_{i,n} &= -\tfrac{1}{r_i} T_{i-1,n} + \tfrac{1}{r_i} A^{-1} T_{i,n} \\ &= -\textstyle\sum_{k=s}^{n} \begin{pmatrix} n \\ k \end{pmatrix} \left(\frac{r_{i-1}^{k+2}}{r_i} s_i^{n-k} F(c.u_k) + \frac{r_{i-1}^{k+1}}{r_i} s_i^{n+1-k} F u_k \right) \\ &\quad + \textstyle\sum_{k=s}^{n} \begin{pmatrix} n \\ k \end{pmatrix} \left(r_i^{k+1} s_i^{n-k} A^{-1}(c.u_k) + r_i^k s_i^{n+1-k} A^{-1} u_k \right). \end{aligned}$$

The result then becomes a consequence of lemmas 4 and 6. □

Lemma 8 *For all n less than $2s-3$*

$$\begin{aligned} FAu_n &= 0, \\ FA(c.A^{-1}u_n) &= 0. \end{aligned}$$

Proof: This follows from the order conditions for the trees $[[\overbrace{\tau,\cdots,\tau}^{n}]]_y$, $[\overbrace{\tau,\cdots,\tau}^{n+1}]_y$ and $[\tau,[\overbrace{\tau,\cdots,\tau}^{n+1}]_z]_y$. □

Lemma 9 *For all n less than $2s-4$*

$$\begin{aligned} F(c^2.u_n) &= 0, \\ F(c.Au_n) &= 0, \\ F(c.A(c.A^{-1}u_n)) &= 0. \end{aligned}$$

Proof: The results follow from the order conditions for the trees $[[\tau,\tau,[\overbrace{\tau,\cdots,\tau}^{n}]_y]_z]_y$, $[[\overbrace{\tau,\cdots,\tau}^{n+3}]_z]_y$, $[[\tau,[[\overbrace{\tau,\cdots,\tau}^{n}]_y]_z]_y$, $[[\tau,[\overbrace{\tau,\cdots,\tau}^{n+1}]_y]_z]_y$ and $[[\tau,[\tau,[\overbrace{\tau,\cdots,\tau}^{n+1}]_z]_y]_z]_y$. □

2.3 Results for the 2-stage method

In order to get a third-order method, w has to satisfy the following linear system $(S_{L,2})$:

$$\begin{aligned} w^T\mathcal{E} &= 1 \quad (0), & w^T\mathcal{C}^2 &= 1 \quad (2), \\ w^T\mathcal{C} &= 1 \quad (1), & w^T\mathcal{A}^{-1}U_2 &= 0 \quad (3). \end{aligned}$$

Taking $\sigma = 2$ leads to a system with 4 equations and 4 unknowns. For convenience, we recall below the coefficients of the 2-stage Radau IIA method,

$$\mathcal{R} = \begin{array}{c|cc} \frac{1}{3} & \frac{5}{12} & -\frac{1}{12} \\ 1 & \frac{3}{4} & -\frac{1}{4} \\ \hline & \frac{3}{4} & -\frac{1}{4} \end{array}.$$

The matrix M corresponding to $(S_{L,2})$ is then of the form

$$M = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{1}{3}r_1 & r_1 & \frac{1}{3}(1+2r_1) & 1 \\ \frac{1}{9}r_1^2 & r_1^2 & \frac{1}{9}(1+2r_1+4r_1^2) & 1 \\ -\frac{2}{27}r_1^2 & \frac{2}{9}r_1^2 & -\frac{2}{27}(1-r_1)^2 & \frac{2}{9}(1-r_1)^2 \end{bmatrix},$$

and we have

$$\det(M) = -\frac{8}{243}r_1(4r_1^3 - 8r_1^2 + 5r_1 - 2).$$

Hence, for all $r_1 \in (0,1)$, M is non-singular and $(S_{L,2})$ has the following unique solution

$$\begin{aligned} w_1 &= \tfrac{3}{2r_1}\tfrac{r_1^5-4r_1^4+7r_1^3-7r_1^2+4r_1-1}{4r_1^3-8r_1^2+5r_1-2}, & w_3 &= \tfrac{1}{2r_1}\tfrac{r_1^5-6r_1^4+11r_1^3-5r_1^2-4r_1+3}{4r_1^3-8r_1^2+5r_1-2}, \\ w_2 &= -\tfrac{3}{2}\tfrac{r_1^4-6r_1^3+12r_1^2-10r_1+3}{4r_1^3-8r_1^2+5r_1-2}, & w_4 &= -\tfrac{1}{2}\tfrac{r_1^4-8r_1^3+12r_1^2-6r_1+3}{4r_1^3-8r_1^2+5r_1-2}. \end{aligned}$$

2.4 Results for the 3-stage method

In order to get a fifth-order method, w must satisfy the following linear system $(S_{L,3})$:

$$\begin{array}{llll} w^T\mathcal{E} = 1 \quad (0), & w^T\mathcal{C}^4 = 1 \quad (5), \\ w^T\mathcal{C} = 1 \quad (1), & w^T\mathcal{A}^{-1}U_4 = 0 \quad (6), \\ w^T\mathcal{C}^2 = 1 \quad (2), & w^T U_3 = 0 \quad (7), \\ w^T\mathcal{C}^3 = 1 \quad (3), & w^T\mathcal{C}.\mathcal{A}^{-1}U_3 = 0 \quad (8), \\ w^T\mathcal{A}^{-1}U_3 = 0 \quad (4), & w^T\mathcal{A}^{-1}(\mathcal{C}.U_3) = 0 \quad (9). \end{array}$$

Taking $\sigma = 3$ now leads to a system with ten equations but only nine unknowns. However, we will show in this section, that one of these equations is identically satisfied. Collecting all results of Section 2.2, we get

$$U_3 = \begin{pmatrix} r_1^4 u_3 \\ r_2^4 u_3 \\ r_3^4 u_3 \end{pmatrix}, \qquad \mathcal{A}^{-1}U_3 = \begin{pmatrix} r_1^3 A^{-1}u_3 \\ r_2^3 A^{-1}u_3 \\ r_3^3 A^{-1}u_3 \end{pmatrix},$$

$$\mathcal{A}^{-1}U_4 = \begin{pmatrix} r_1^4 A^{-1}u_4 \\ r_2^4 A^{-1}u_4 + 4r_2^3 s_2 A^{-1}u_3 \\ r_3^4 A^{-1}u_4 + 4r_3^3 s_3 A^{-1}u_3 \end{pmatrix}, \qquad \mathcal{A}^{-1}(\mathcal{C}.U_3) = \begin{pmatrix} r_1^4 A^{-1}(c.u_3) \\ r_2^4 A^{-1}(c.u_3) + r_2^3 s_2 A^{-1}u_3 \\ r_3^4 A^{-1}(c.u_3) + r_3^3 s_3 A^{-1}u_3 \end{pmatrix},$$

$$\mathcal{C}.\mathcal{A}^{-1}U_3 = \begin{pmatrix} r_1^4 c.A^{-1}u_3 \\ r_2^4 c.A^{-1}u_3 + r_2^3 s_2 A^{-1}u_3 \\ r_3^4 c.A^{-1}u_3 + r_3^3 s_3 A^{-1}u_3 \end{pmatrix}.$$

Let $\mathcal{V}_1$ be $\mathcal{A}^{-1}U_4 - 4\mathcal{A}^{-1}(\mathcal{C}.U_3)$, $\mathcal{V}_2$ be $\mathcal{C}.\mathcal{A}^{-1}U_3 - \mathcal{A}^{-1}(\mathcal{C}.U_3)$, v_1 be $A^{-1}u_4 - 4A^{-1}(c.u_3)$ and v_2 be $c.A^{-1}u_3 - A^{-1}(c.u_3)$. It is found that

$$\mathcal{V}_1 = \begin{pmatrix} r_1^4 v_1 \\ r_2^4 v_1 \\ r_3^4 v_1 \end{pmatrix} \text{ and } \mathcal{V}_2 = \begin{pmatrix} r_1^4 v_2 \\ r_2^4 v_2 \\ r_3^4 v_2 \end{pmatrix},$$

and the system $(S_{L,3})$ is equivalent to

$$\begin{array}{llll} w^T\mathcal{E} = 1 \quad (0), & w^T\mathcal{C}^4 = 1 \quad (5), \\ w^T\mathcal{C} = 1 \quad (1), & w^T\mathcal{V}_1 = 0 \quad (6), \\ w^T\mathcal{C}^2 = 1 \quad (2), & w^T U_3 = 0 \quad (7), \\ w^T\mathcal{C}^3 = 1 \quad (3), & w^T\mathcal{V}_2 = 0 \quad (8), \\ w^T\mathcal{A}^{-1}U_3 = 0 \quad (4), & w^T\mathcal{A}^{-1}(\mathcal{C}.U_3) = 0 \quad (9). \end{array}$$

Theorem 1 *$\mathcal{V}_1$, $\mathcal{V}_2$ and U_3 are linearly dependent.*

Proof: It is enough to show that v_1, v_2 and u_3 are linearly dependent. Since $\mathcal{R} = (A, b, c)$ is of order 5 for the differential component, we have

$$\begin{array}{llll} b^T A^{-1}c^5 = 1, & b^T A^{-1}(c.Ac^3) = \frac{1}{4}, \\ b^T Ac^3 = \frac{1}{20}, & b^T c^4 = \frac{1}{5}, \\ b^T c.A^{-1}c^4 = \frac{4}{5}, \end{array}$$

so that $b^T u_3 = b^T v_1 = b^T v_2 = 0$. Now, $b \neq 0$ implies the result.□

Remark 1 *Whenever $r_1 = r_2 = r_3 = \frac{1}{3}$, the vectors $\mathcal{A}^{-1}U_3$, U_3 and $\mathcal{V}_1$ (for example) become linearly dependent. To prove this, it is sufficient to show that $A^{-1}u_3$, u_3 and v_1 are dependent. As $b^T A^{-1}u_3 =$*

$b^T c^3 - \frac{1}{4} b^T A^{-1} c^4 = 0$, we can conclude as in Theorem 1. In this case, w depends on one parameter that can be chosen so as to minimize the quantity

$$\sum_{u \in DAT2_z(5)} \alpha(u) \| \gamma(u) w^T \Phi(u) - 1 \|,$$

where $DAT2_z(5) = \{u \in DAT2_z \mid \rho(u) = 5\}$. This seems a natural goal to achieve, since this is an attempt to minimize the local error. For convenience, Table 2 collects the trees of $DAT2_z(5)$ and the values of the associated functions α, γ and Φ. To compute the α's, we refer to [Hig93].

tree u	$\alpha(u)$	$\gamma(u)$	$\Phi(u)$
$[[\overbrace{\tau, \cdots, \tau}^{5}]_y]_z$	1	1	c^5
$[\overbrace{\tau, \cdots, \tau}^{6}]_z$	1	$\frac{1}{6}$	$A^{-1}c^6$
$[[[\tau, \tau, \tau, \tau]_y]_y]_z$	1	5	Ac^4
$[[\tau, [\overbrace{\tau, \cdots, \tau}^{5}]_z]_y]_z$	5	$\frac{1}{5}$	$c.A^{-1}c^5$
$[\tau, [\tau, \tau, \tau, \tau]_y]_z$	6	$\frac{5}{6}$	$A^{-1}(c.Ac^4)$
$[\tau, [[\tau, \tau, \tau]_y]_y]_z$	6	$\frac{10}{3}$	$A^{-1}(c.A^2c^3)$
$[\tau, [\tau, [\tau, \tau, \tau]_z]_y]_z$	24	$\frac{5}{24}$	$A^{-1}(c.A(c.A^{-1}c^4))$
$[\tau, \tau, [\tau, \tau, \tau]_y]_z$	15	$\frac{2}{3}$	$A^{-1}(c^2.Ac^3)$
$[[[[\tau, \tau, \tau]_y]_y]_y]_z$	1	20	A^2c^3
$[[[\tau, [\tau, \tau, \tau, \tau]_z]_y]_y]_z$	4	$\frac{5}{4}$	$A(c.A^{-1}c^4)$
$[[\tau, [\tau, \tau, \tau]_y]_y]_z$	5	4	$c.Ac^3$
$[[\tau, [\tau, [\tau, \tau, \tau]_y]_z]_y]_z$	25	$\frac{4}{5}$	$c.A^{-1}(c.Ac^3)$
$[[\tau, \tau, [\tau, \tau, \tau, \tau]_z]_y]_z$	10	$\frac{1}{4}$	$c^2.A^{-1}c^4$

Table 2: Trees of $DAT2_z(5)$ and their associated functions

2.5 Sketch of the case $s = 4$

To achieve order 7 for the algebraic component, we have to solve the system $(S_{L,4})$ composed of (C_4), (C_5), (C_6) and (C_7) (see Section 2.1), that is to say 24 equations. Comparing the number of equations and the number of unknown, we could consequently think of taking $\sigma = 6$. In fact, it can be shown that $\sigma = 5$ is sufficient, since 5 equations are identically satisfied (see Section 6.1). However, it does not seem reasonable any more to consider a practical implementation of the corresponding method, owing to the complexity of the formulas for variable stepsize.

Remark 2 *If the stepsize is constant (i.e. $r_1 = r_2 = \cdots = r_p$) then 9 equations are identically satisfied (see Section 6.1), and σ can be chosen equal to* 4.

3 Modifications to the code RADAU5

The 3-stage Radau IIA method has been implemented by E. Hairer and G. Wanner in order to solve problems of the form $MY' = F(Y)$. ODEs and DAEs of index less than or equal to three can be solved by this code, called RADAU5. A precise description is given in Section IV-8 [HW91] and we will adopt the notations used there. Implementing our method requires slight modifications to the subroutines "radcor" and "solout" which are actually replaced respectively by "radcorz" and "soloutz".

3.1 Modifications to radcor

Only the computation of the algebraic component (z) is modified (if an index 2 DAE is solved). Once the n^{th} step has been accepted, two cases are considered:

1. less than three steps have been computed. Then, we keep the internal stages $z_{n,1}$, $z_{n,2}$, $z_{n,3}$ and the step size h_n. The value of (y_{n+1}, z_{n+1}) is $(y_{n,3}, z_{n,3})$.

2. three or more steps have been computed. Then, we keep the internal stages $z_{n,1}$, $z_{n,2}$, $z_{n,3}$ and the step size h_n. r_1, r_2 are computed by the formulas

$$r_1 = \frac{h_{n-2}}{h_{n-2} + h_{n-1} + h_n} \text{ and } r_2 = \frac{h_{n-1}}{h_{n-2} + h_{n-1} + h_n},$$

and w by the subroutine "vectw2". For (y_{n+1}, z_{n+1}) we put

$$\begin{aligned} y_{n+1} &= y_{n,3}, \\ z_{n+1} &= w_1 z_{n-2,1} + w_2 z_{n-2,2} + w_3 z_{n-2,3} + w_4 z_{n-1,1} + w_5 z_{n-1,2} + \\ &\quad w_6 z_{n-1,3} + w_7 z_{n,1} + w_8 z_{n,2} + w_9 z_{n,3}. \end{aligned}$$

Remark 3 *For continuous outputs, we need also to keep the internal stages over two steps for the differential components (see section 3.2.2).*

3.2 New subroutines

3.2.1 Vectw2

In order to compute the formal expression of w and to create the associated fortran subroutine, the manipulation package Maple was used. A call to vectw2 uses the format VECTW2(ICAS,VW,R1,R2), where the inputs are one of the five cases described below (ICAS) and the parameters r_1, r_2 (R1,R2). The output is the vector w (VW). Five cases are considered:

1. $r_1 = r_2 = r_3 = \frac{1}{3}$. In this case, we have seen that w depends on one parameter. It is optimized as explained in Remark 1.

2. $r_1 = r_2$ and $r_3 \neq r_1$.

3. $r_2 = r_3$ and $r_1 \neq r_2$.

4. $r_1 = r_3$ and $r_2 \neq r_1$.

5. $r_1 \neq r_2$, $r_1 \neq r_3$ and $r_2 \neq r_3$.

This allows us to reduce the cost of computation and to eliminate computational problems: had we used the general expression of w (case 5), divisions by zero would have occured in the cases 1 to 4.

3.2.2 Continuous outputs

In the code Radau5, the subroutine "solout" provides the user with approximations at equidistant output-points. The corresponding interpolation formulas are implemented in the subroutine "contr5". "contr5(I,x)" gives an approximation $U^I(X)$ to the I^{th} component of the solution Y at the point x (x should lie in the interval $[x_n, x_{n+1}]$). U is the collocation polynomial: it is of degree 3 and defined by

$$\begin{aligned} U(x_n) &= Y_n, \\ U(x_n + c_i h_n) &= Y_{n,i}, \quad i = 1, 2, 3. \end{aligned}$$

For index 2 DAEs, $U = (u, v)$ where u, v are polynomials of degree 3 which satisfy

$$\begin{array}{rclrcl} u(x_n) & = & y_n, & v(x_n) & = & z_n, \\ u(x_n + c_i h_n) & = & y_{n,i}, & v(x_n + c_i h_n) & = & z_{n,i}, \quad i = 1, 2, 3. \end{array}$$

By Theorem 7-8 ([HW91]), we have

$$u(x) - y(x) = O(h^4) \text{ and } v(x) - z(x) = O(h^3).$$

As our aim is to increase the order of convergence for the algebraic component, it seems natural to search for an approximation $P(x) = (p(x), q(x))$ of $Y(x) = (y(x), z(x))$ satisfying $P(x) - Y(x) = O(h^5)$.

Approximation of $z(x)$
Let x be of the form $x_{n-2} + \theta h$ where $h = h_{n-2} + h_{n-1} + h_n$. We define q as follows

$$q(x) = \sum_{i=1}^{3} w_i(\theta) z_{n-2,i} + w_{3+i}(\theta) z_{n-1,i} + w_{6+i}(\theta) z_{n,i},$$

where the vector $w(\theta) = (w_1(\theta), \cdots, w_9(\theta))^T$ satisfies the linear system

$$S^z_{L,3}(\theta) \left\{ \begin{array}{llll} w^T(\theta)\mathcal{E} = 1 & (0), & w^T(\theta)\mathcal{C}^4 = \theta^4 & (5), \\ w^T(\theta)\mathcal{C} = \theta & (1), & w^T(\theta)\mathcal{V}_1 = 0 & (6), \\ w^T(\theta)\mathcal{C}^2 = \theta^2 & (2), & w^T(\theta)U_3 = 0 & (7), \\ w^T(\theta)\mathcal{C}^3 = \theta^3 & (3), & w^T(\theta)\mathcal{V}_2 = 0 & (8), \\ w^T(\theta)\mathcal{A}^{-1}U_3 = 0 & (4), & w^T(\theta)\mathcal{A}^{-1}(\mathcal{C}.U_3) = 0 & (9). \end{array} \right.$$

Using the notations of section 2.4, we have

Proposition 2 *For all $\theta \in (0, 1]$, $S^z_{L,3}(\theta)$ possesses a solution and*

$$q(x_{n-2} + \theta h) - z(x_{n-2} + \theta h) = O(h^5).$$

Proof: From Theorem 8-5 and 8-6 in [HW91], we have

$$q(x_{n-2} + \theta h) - z(x_{n-2} + \theta h) = O(h^5) \text{ iff } w^T(\theta)\Phi(u) = \frac{\theta^{\rho(u)}}{\gamma(u)}, \ \forall u \in DAT2_z, \ \rho(u) \leq 4.$$

According to the analysis of section 2.1, this leads to the system $S^z_{L,3}(\theta)$ which, by Theorem 1, possesses a solution. □

The subroutine "vectwz" computes $w(\theta)$. As for the vector w, five cases are considered. When $h_{n-2} = h_{n-1} = h_n$, $w(\theta)$ depends on one parameter (case 1) which is not adjusted as in Remark 1. In this case, the value defined by continuity for $w(\theta)$ is choosen. A call to "vectwz" uses the format VECTWZ(ICAS,VWZ,R1,R2,T), where the inputs are one of the five cases described before (ICAS) and the parameters r_1, r_2, θ (respectively R1,R2,T). The output is the vector $w(\theta)$ (VWZ).

Approximation of $y(x)$
Let x be of the form $x_{n-1} + \eta h$ where $h = h_{n-1} + h_n$ and $h_{n-1} = rh$. We define p as follows

$$p(x) = \sum_{i=1}^{3} B_i(\eta) z_{n-1,i} + B_{3+i}(\eta) z_{n,i},$$

where the vector $B(\eta) = (B_1(\eta), \cdots, B_6(\eta))^T$ satisfies the linear system

$$S^y_{L,3}(\eta) \left\{ \begin{array}{llllllll} B^T(\eta)E & = & 1 & (0), & B^T(\eta)C^3 & = & \eta^3 & (3), \\ B^T(\eta)C & = & \eta & (1), & B^T(\eta)C^4 & = & \eta^4 & (4), \\ B^T(\eta)C^2 & = & \eta^2 & (2), & B^T(\eta)\mathcal{A}C^3 & = & \frac{\eta^4}{4} & (5). \end{array} \right.$$

$(\mathcal{A}, \mathcal{A}^T B(1), C)$ is the Runge-Kutta method $\mathcal{R}_2$ obtained by the composition of the 3-stage Radau IIA method $\mathcal{R} = (A, b, c)$ over two steps,

$$\mathcal{R}_2 = \begin{array}{c|cc} rc & rA & 0 \\ re + (1-r)c & reb^T & (1-r)A \\ \hline & rb & (1-r)b \end{array},$$

and E is the vector $(\overbrace{1, \cdots, 1}^{6})^T$.

Proposition 3

$$\forall \eta \in (0, 1],\ p(x_{n-1} + \eta h) - y(x_{n-1} + \eta h) = O(h^5).$$

Proof: Let us introduce the vector $D(\eta) = \mathcal{A}^{-T} B(\eta) = (D_i(\eta))_{i=1,\cdots,6}$ and the following polynomial u:

$$u(x_{n-1} + \eta h) = y_{n-1} + \sum_{i=1}^{3} D_i(\eta) f(y_{n-1,i}, z_{n-1,i}) + D_{i+3}(\eta) f(y_{n,i}, z_{n,i}).$$

From Theorem 8-5 and 8-6 in [HW91], we have

$$u(x_{n-1} + \eta h) - y(x_{n-1} + \eta h) = O(h^5) \text{ iff } D^T(\eta)\Phi(t) = \frac{\eta^{\rho(t)}}{\gamma(t)}\ \forall t \in DAT2_y,\ \rho(t) \leq 4\ \ (P).$$

Using the points of the internal stages, it follows

$$u(x_{n-1} + \eta h) = R(\infty) y_{n-1} + \sum_{i=1}^{3} D_i(\eta) y_{n-1,i} + D_{i+3}(\eta) y_{n,i},$$

so that (P) is equivalent to the system $S^y_{L,3}(\eta)$. □

The subroutine "vectwy" computes $B(\eta)$. A call to "vectwy" uses the format VECTWY(VWY,R,T), where the inputs are the parameters r and η (respectively R,T) and the output the vector $B(\eta)$ (VWY).

The subroutine "soloutz" provides the user with approximations $(p(x^i_{out}), q(x^i_{out}))$ of $(y(x^i_{out}), z(x^i_{out}))$ at equidistant output-points $(x^i_{out})_{i=1,\cdots,N}$. For the differential component, x^i_{out} is of the form $x_{n-1} + \eta(h_{n-1} + h_n)$ where η is choosen so as to satisfy $x_n < x^i_{out} \leq x_{n+1}$ and for the algebraic one, x^i_{out} is of the form $x_{n-2} + \theta(h_{n-2} + h_{n-1} + h_n)$ where θ is choosen in satisfy $x_{n-1} < x^i_{out} \leq x_n$. A call to "soloutz" uses the format

SOLOUTZ(NR,XOLD,X,Y,NEQN,ICAS,R1,R2,HTOT,H2,H3,LAST),

where the inputs are the number of accepted steps (NR), x_n (XOLD), x_{n+1} (X), (y_{n+1}, z_{n+1}) (Y), the system's dimension (NEQ), one of the five cases described before for the computation of $w(\theta)$ (ICAS), the parameters $r1$, $r2$ (R1,R2), the stepsize $h_{n-2} + h_{n-1} + h_n$, h_{n-1}, h_n (HTOT,H2,H3) and a Boolean variable (LAST) to indicate if the last computational step has been reached.

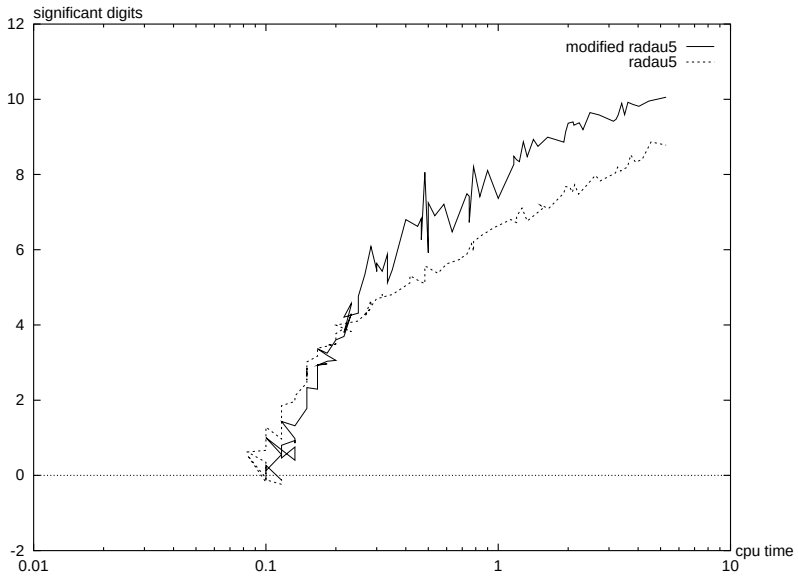

Figure 1: Precision versus computing time - algebraic components - test problem

4 Numerical experiments

4.1 Test problem

We consider the index two problem:

$$\begin{cases} y'(t) &= \Psi'(t) \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} y(t) + z(t)y(t) \\ 0 &= y^T(t)y(t) - 1 \end{cases} , t \in [-1, 11],$$

where Ψ is the following infinitely smooth function:

$$\Psi(t) = \begin{cases} \frac{\pi}{2}\exp\left(\frac{t^2}{t^2-1}\right) & \text{if } |t| < 1, \\ \frac{\pi}{2}\exp\left(\frac{(t-5)^2}{(t-5)^2-1}\right) & \text{if } |t-5| < 1, \\ \frac{\pi}{2}\exp\left(\frac{(t-10)^2}{(t-10)^2-1}\right) & \text{if } |t-10| < 1, \\ 0 & \text{otherwise,} \end{cases}$$

with consistant initial values. The exact solution is

$$\begin{aligned} y_1(t) &= \cos(\Psi(t)), \\ y_2(t) &= \sin(\Psi(t)), \\ z(t) &= 0. \end{aligned}$$

In both codes, we set WORK(4) $= 0.001$, WORK(5) $= 0.99$, WORK(6) $= 1.3$ and initial step size $h = 10^{-7}$ (WORK(4) is the parameter κ in the stopping criterion for Newton's method. WORK(4), WORK(5) are the parameters c_1, c_2 in the stepsize control, see [HW91], page 130-134).
In figure 1, we plot the CPU time against the number of significant digits ($-\log_{10}$ (absolute error)) of the algebraic components, for both codes. For this, we use continuous outputs : outputs are required at $t_i = t_{i-1} + 0.2$, we compute the global error and then take the maximum over all values. In figure 2, we plot the CPU time against the number of significant figures of the differential components, for both codes.
In the following problems, only the modified parameters will be shown.

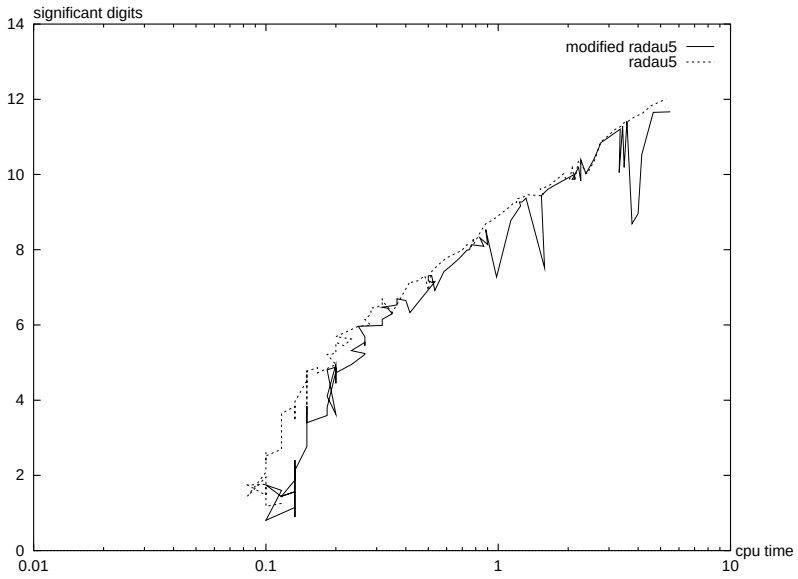

Figure 2: Precision versus computing time - differential components - test problem

4.2 Pendulum

The simplest constrained mechanical system is the pendulum, whose equations of motions are described in [HW91], page 483-485. We have applied the code Radau5 and the modified code to the GGL (Gear, Gupta and Leimkuhler) formulation

$$\left\{\begin{array}{lcll} p'(t) & = & u(t) - p(t)\mu(t), & \\ q'(t) & = & v(t) - q(t)\mu(t), & \\ mu'(t) & = & -p(t)\lambda(t), & \\ mv'(t) & = & -q(t)\lambda(t) - g, & t \in [0, 10], \\ 0 & = & p(t)^2 + q(t)^2 - l^2, & \\ 0 & = & p(t)u(t) + q(t)v(t), & \end{array}\right.$$

with consistant initial values $p(0) = 1$, $q(0) = 0$, $u(0) = 0$, $v(0) = 0$, $\lambda(0) = 0$ and $\mu(0) = 0$. For simplicity, we took $m = 1$, $g = 1$ and $l = 1$.
In figure 3 (respectively 4), we plot the CPU time against the number of significant digits of the algebraic (resp. differential) components, for both codes.

4.3 Multibody mechanism

A seven body mechanism is described in [HW91], page 531-545. We have applied the code Radau5 and the modified code to the index 2 formulation

$$\left\{\begin{array}{lcl} q'(t) & = & v(t), \\ v'(t) & = & M(q(t))^{-1}\left(f(q(t), v(t)) - G^T(q(t))\lambda(t)\right), \quad t \in [0, 3.10^{-2}], \\ 0 & = & G(q(t))v(t), \end{array}\right.$$

with consistant initial values.
In figure 5 (respectively 6), we plot the CPU time against the number of significant digits of the algebraic (resp. differential) components, for both codes. Here, outputs are required at $t_i = t_{i-1} + 0.0003$.

4.4 Discharge pressure control

This simplified model of a dynamic simulation problem in petrochimical engineering is described in [HLR89], page 116-118. We have applied the code Radau5 and the modified code to the following

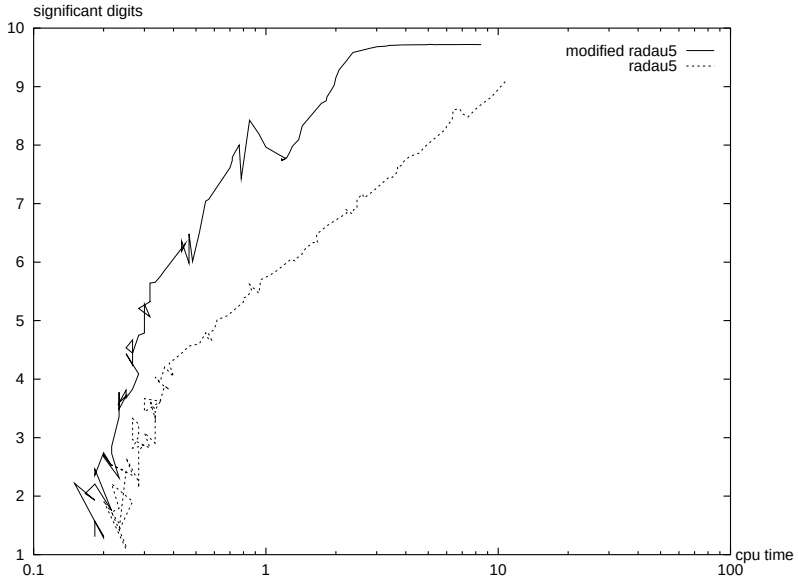

Figure 3: Precision versus computing time - algebraic components - pendulum

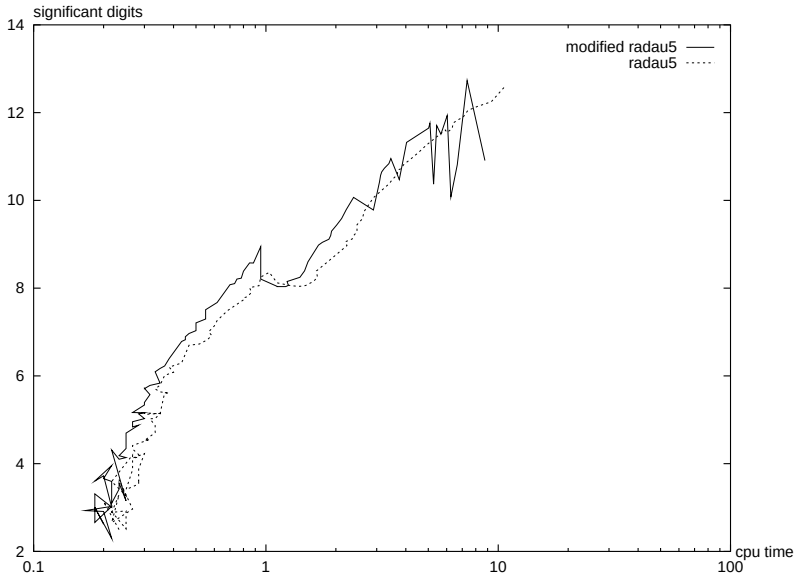

Figure 4: Precision versus computing time - differential components - pendulum

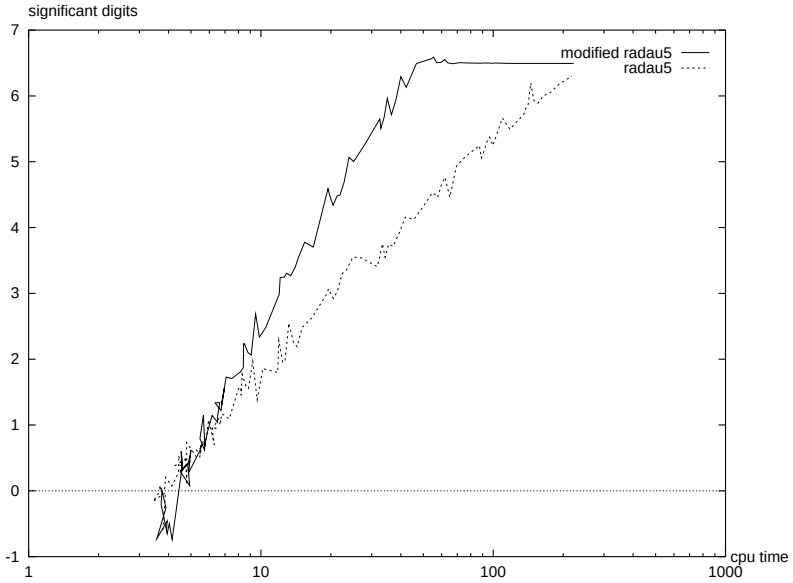

Figure 5: Precision versus computing time - algebraic components - seven body mechanism

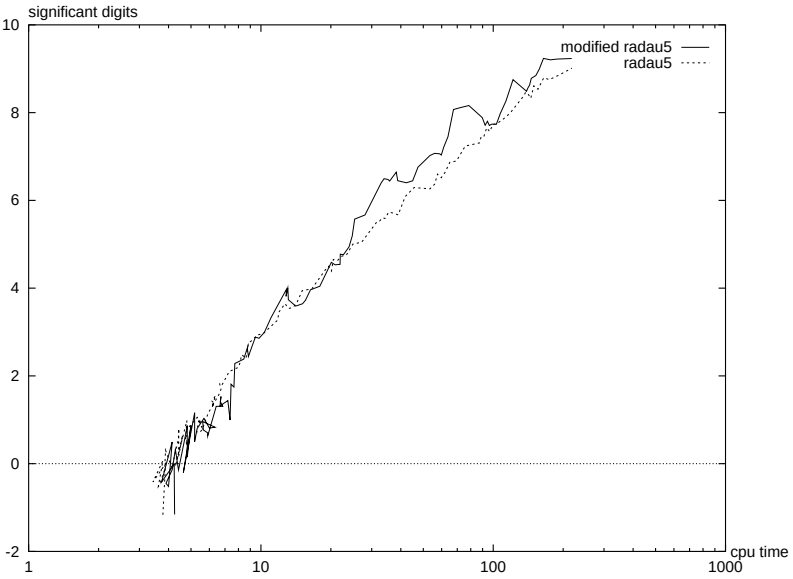

Figure 6: Precision versus computing time - differential components - seven body mechanism

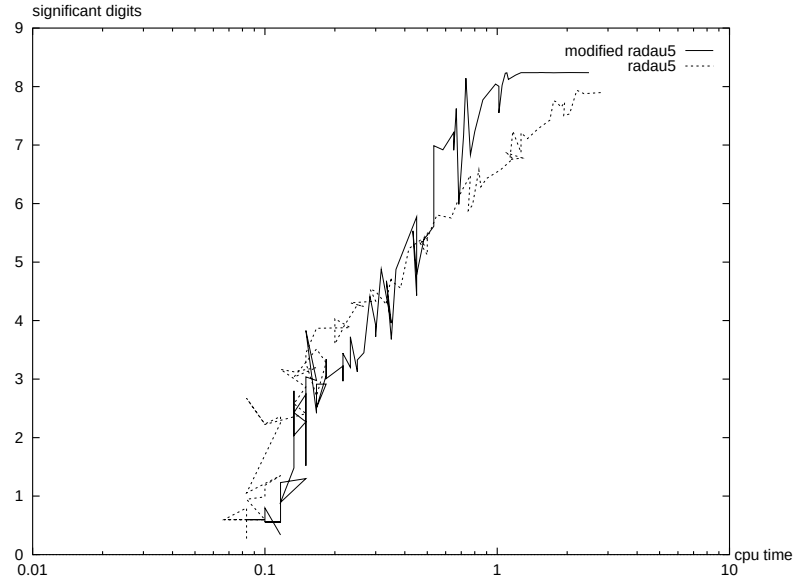

Figure 7: Precision versus computing time - algebraic components - discharge pressure control

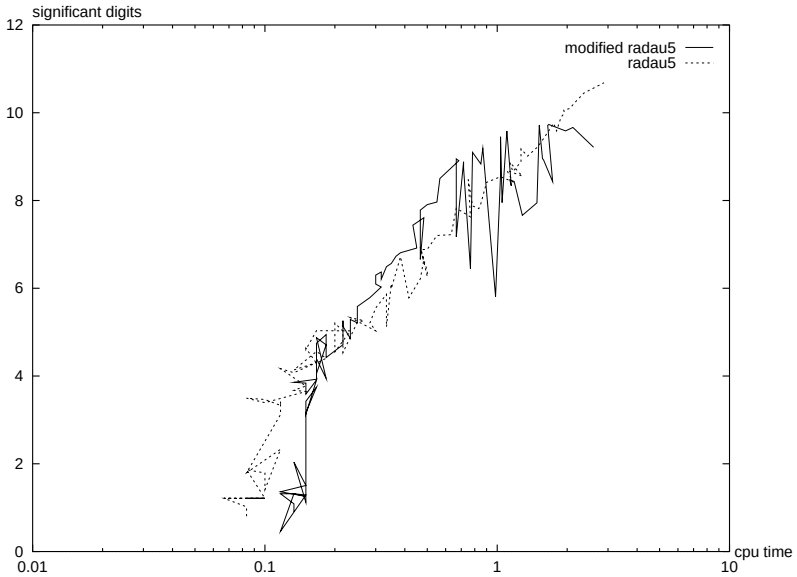

Figure 8: Precision versus computing time - differential components - discharge pressure control

index 2 formulation:

$$\left\{\begin{array}{lcll} \mu(t) & = & 15+\tanh(t-10), & \\ p(t) & = & \frac{M}{20}(3.35-0.075m(t)+0.001m(t)^2), & \\ k'(t) & = & \frac{1}{20}\left(s(t)-\frac{p(t)}{15}-k(t)\right), & \\ s'(t) & = & -\frac{1}{75}(p(t)-99.1), & t\in[0,40], \\ M'(t) & = & \mu(t)-m(t), & \\ 0 & = & \left(\frac{M(t)}{20}\right)^2-49.58^2+\left(\frac{\mu(t)}{1.2k(t)}\right)^2, & \end{array}\right.$$

with consistant initial values. Here, initial step size h is equal to 10^{-5}.
In figure 7 (respectively 8), we plot the CPU time against the number of significant digits of the algebraic (resp. differential) components, for both codes. Here, outputs are required at $t_i = t_{i-1} + 0.5$.

5 Conclusion

A new simple technique to overcome the order reduction phenomenon, appearing for the algebraic component when Radau IIA methods are applied to index two DAEs, is proposed.
Increasing the order of convergence of z is made possible by considering the composition of the basic method with itself over σ steps. As z_{n+1} is defined in the basic method as a linear combination of the internal stages, a good choice of σ should provide enough freedom for the order conditions associated with the composite method to be satisfied. We have determined these order conditions which derive straightforwardly from the work of E. Hairer, C. Lubich and M. Roche (section 2.1). Then we have shown that some of those conditions are redundant and might be omitted for s-stage Radau IIA methods with $s \leq 4$ (section 2.3 to 2.5).

It could be interesting to generalize this simplifications to any Radau IIA methods. A general question will be to determine how many compositions of a s-stage Radau IIA method have to be considered to obtain an order of convergence equal to $2s-1$ for the algebraic component. However, it does not seem reasonable any more to consider a practical implementation of the corresponding method for $s \geq 4$, owing to the complexity of the formulas for variable stepsize.

The formulas for $s = 3$ have been incorporated in the code Radau5 developed by E. Hairer and G. Wanner. Slight modifications were required (section 3). Only the computation of the algebraic component was modified and a new procedure in order to have continuous outputs was created where we have used our technique for both components (algebraic and differential) to compute approximations of order five at equidistant output-points.

According to our numerical experiments, results for the differential components are disappointing. However, the use of our technique in the code Radau5 leads to an increase of the accuracy for the algebraic components (when tolerances are sufficiently small).

6 Annexe

6.1 Construction of the vector w in the case $s = 4$

In this section, we explained the calculus of linear algebra used to show that composed five times the s-stage Radau IIA method is sufficient in the case $s = 4$. Let expand the following vectors of (C_6) (We use Lemma 8)

$$\begin{aligned}
\mathcal{A}^{-1}U_4 &= \left[\left(r_i^4 A^{-1}u_4\right)^T\right]^T_{i=1,\cdots,\sigma},\\
\mathcal{A}^{-1}U_5 &= \left[\left(r_i^5 A^{-1}u_5 + 5r_i^4 s_i A^{-1}u_4\right)^T\right]^T_{i=1,\cdots,\sigma},\\
U_4 &= \left[\left(r_i^5 u_4\right)^T\right]^T_{i=1,\cdots,\sigma},\\
\mathcal{C}.\mathcal{A}^{-1}U_4 &= \left[\left(r_i^5 c.A^{-1}u_4 + r_i^4 s_i A^{-1}u_4\right)^T\right]^T_{i=1,\cdots,\sigma},\\
\mathcal{A}^{-1}(\mathcal{C}.U_4) &= \left[\left(r_i^5 A^{-1}(c.u_4) + r_i^4 s_i A^{-1}u_4\right)^T\right]^T_{i=1,\cdots,\sigma},
\end{aligned}$$

and the following vectors of (C_7) (We use Lemma 9)

$$
\begin{aligned}
\mathcal{A}^{-1}U_6 &= \left[\left(r_i^6 A^{-1}u_6 + 6r_i^5 s_i A^{-1}u_5 + 5r_i^4 s_i^2 A^{-1}u_4\right)^T\right]^T_{i=1,\cdots,\sigma},\\
U_5 &= \left[\left(r_i^6 u_5 + 5r_i^5 s_i u_4\right)^T\right]^T_{i=1,\cdots,\sigma},\\
\mathcal{C}.\mathcal{A}^{-1}U_5 &= \left[\left(r_i^6 c.A^{-1}u_5 + r_i^5 s_i A^{-1}u_5 + 5r_i^5 s_i c.A^{-1}u_4 + 5r_i^4 s_i^2 A^{-1}u_4\right)^T\right]^T_{i=1,\cdots,\sigma},\\
\mathcal{A}^{-1}(\mathcal{C}.U_5) &= \left[\left(r_i^6 A^{-1}(c.u_5) + r_i^5 s_i A^{-1}u_5 + 5r_i^5 s_i A^{-1}(c.u_4) + 5r_i^4 s_i^2 A^{-1}u_4\right)^T\right]^T_{i=1,\cdots,\sigma},\\
\mathcal{A}^{-1}(\mathcal{C}.\mathcal{A}U_4) &= \left[\left(r_i^6 A^{-1}(c.Au_4) + r_i^5 s_i u_4\right)^T\right]^T_{i=1,\cdots,\sigma},\\
\mathcal{A}^{-1}(\mathcal{C}.\mathcal{A}(\mathcal{C}.\mathcal{A}^{-1}U_4)) &= [(r_i^6 A^{-1}(c.A(c.A^{-1}u_4)) + r_i^5 s_i A^{-1}(c.u_4) + r_i^5 s_i c.A^{-1}u_4\\
&\quad + r_i^4 s_i^2 A^{-1}u_4)^T\Big]^T_{i=1,\cdots,\sigma},\\
\mathcal{A}^{-1}(\mathcal{C}^2.U_4) &= \left[\left(r_i^6 A^{-1}(c^2.u_4) + 2r_i^5 s_i A^{-1}(c.u_4) + r_i^4 s_i^2 A^{-1}u_4\right)^T\right]^T_{i=1,\cdots,\sigma},\\
\mathcal{A}U_4 &= \left[\left(r_i^6 Au_4\right)^T\right]^T_{i=1,\cdots,\sigma},\\
\mathcal{A}(\mathcal{C}.\mathcal{A}^{-1}U_4) &= \left[\left(r_i^6 A(c.A^{-1}u_4) + r_i^5 s_i u_4\right)^T\right]^T_{i=1,\cdots,\sigma},\\
\mathcal{C}.U_4 &= \left[\left(r_i^6 c.u_4 + r_i^5 s_i u_4\right)^T\right]^T_{i=1,\cdots,\sigma},\\
\mathcal{C}.\mathcal{A}^{-1}(\mathcal{C}.U_4) &= \left[\left(r_i^6 c.A^{-1}(c.u_4) + r_i^5 s_i c.A^{-1}u_4 + r_i^5 s_i A^{-1}(c.u_4) + r_i^4 s_i^2 A^{-1}u_4\right)^T\right]^T_{i=1,\cdots,\sigma},\\
\mathcal{C}^2.\mathcal{A}^{-1}U_4 &= \left[\left(r_i^6 c^2.A^{-1}u_4 + 2r_i^5 s_i c.A^{-1}u_4 + r_i^4 s_i^2 A^{-1}u_4\right)^T\right]^T_{i=1,\cdots,\sigma}.
\end{aligned}
$$

Let introduce the following vectors

$$
\begin{aligned}
\mathcal{T}_1 &= \mathcal{A}^{-1}U_5 - 5\mathcal{A}^{-1}(\mathcal{C}.U_4),\\
\mathcal{T}_2 &= \mathcal{C}.\mathcal{A}^{-1}U_4 - \mathcal{A}^{-1}(\mathcal{C}.U_4),\\
\mathcal{V}_1 &= U_5 - 5\mathcal{A}(\mathcal{C}.\mathcal{A}^{-1}U_4),\\
\mathcal{V}_2 &= U_5 - 5\mathcal{A}^{-1}(\mathcal{C}.\mathcal{A}U_4),\\
\mathcal{V}_3 &= U_5 - 5\mathcal{C}.U_4,\\
\mathcal{V}_4 &= \mathcal{C}.\mathcal{A}^{-1}(\mathcal{C}.U_4) - \mathcal{A}^{-1}(\mathcal{C}.\mathcal{A}(\mathcal{C}.\mathcal{A}^{-1}U_4)),\\
\mathcal{V}_5 &= \mathcal{A}^{-1}(\mathcal{C}^2.U_4) + \mathcal{C}^2.\mathcal{A}^{-1}U_4 - \mathcal{C}.\mathcal{A}(\mathcal{C}.\mathcal{A}^{-1}U_4) - \mathcal{C}.\mathcal{A}^{-1}(\mathcal{C}.U_4),\\
\mathcal{V}_6 &= 2\left(\mathcal{C}.\mathcal{A}^{-1}U_5 - \mathcal{A}^{-1}(\mathcal{C}.U_5)\right) - 5\left(\mathcal{C}^2.\mathcal{A}^{-1}U_4 - \mathcal{A}^{-1}(\mathcal{C}^2.U_4)\right),\\
\mathcal{V}_7 &= 15\left(\mathcal{C}^2.\mathcal{A}^{-1}U_4 + \mathcal{A}^{-1}(\mathcal{C}^2.U_4)\right) - 6\left(\mathcal{C}.\mathcal{A}^{-1}U_5 + \mathcal{A}^{-1}(\mathcal{C}.U_5)\right) + 2\mathcal{A}^{-1}U_6,
\end{aligned}
$$

(idem for the vectors $t_1 = A^{-1}u_5 - 5A^{-1}c.u_4$, t_2, v_1, ... ,v_7) then

$$
\begin{aligned}
\mathcal{T}_1 &= \left[r_i^5\left(A^{-1}u_5 - 5A^{-1}(c.u_4)\right)^T\right]^T_{i=1,\cdots,\sigma},\\
\mathcal{T}_2 &= \left[r_i^5\left(c.A^{-1}u_4 - A^{-1}(c.u_4)\right)^T\right]^T_{i=1,\cdots,\sigma},\\
\mathcal{V}_1 &= \left[r_i^6\left(u_5 - 5A(c.A^{-1}u_4)\right)^T\right]^T_{i=1,\cdots,\sigma},\\
\mathcal{V}_2 &= \left[r_i^6\left(u_5 - 5A^{-1}(c.Au_4)\right)^T\right]^T_{i=1,\cdots,\sigma},\\
\mathcal{V}_3 &= \left[r_i^6\left(u_5 - 5c.u_4\right)^T\right]^T_{i=1,\cdots,\sigma},\\
\mathcal{V}_4 &= \left[r_i^6\left(c.A^{-1}(c.u_4) - A^{-1}(c.A(c.A^{-1}u_4))\right)^T\right]^T_{i=1,\cdots,\sigma},\\
\mathcal{V}_5 &= \left[r_i^6\left(A^{-1}(c^2.u_4) + c^2.A^{-1}u_4 - c.A(c.A^{-1}u_4) - c.A^{-1}(c.u_4)\right)^T\right]^T_{i=1,\cdots,\sigma},\\
\mathcal{V}_6 &= \left[r_i^6\left(2\left(c.A^{-1}u_5 - A^{-1}(c.u_5)\right) - 5\left(c^2.A^{-1}u_4 - A^{-1}(c^2.u_4)\right)\right)^T\right]^T_{i=1,\cdots,\sigma},\\
\mathcal{V}_7 &= \left[r_i^6\left(-6\left(c.A^{-1}u_5 + A^{-1}(c.u_5)\right) + 15\left(c^2.A^{-1}u_4 + A^{-1}(c^2.u_4)\right) + 2A^{-1}u_6\right)^T\right]^T_{i=1,\cdots,\sigma},
\end{aligned}
$$

and the system $(S_{L,4})$ is equivalent to

$$
\begin{array}{llll}
w^T\mathcal{E} = 1 \ (0), & w^T\mathcal{A}^{-1}U_6 = 0 \ (12), \\
w^T\mathcal{C} = 1 \ (1), & w^TU_5 = 0 \ (13), \\
w^T\mathcal{C}^2 = 1 \ (2), & w^T\mathcal{V}_6 = 0 \ (14), \\
w^T\mathcal{C}^3 = 1 \ (3), & w^T\mathcal{V}_7 = 0 \ (15), \\
w^T\mathcal{C}^4 = 1 \ (4), & w^T\mathcal{V}_2 = 0 \ (16), \\
w^T\mathcal{A}^{-1}U_4 = 0 \ (5), & w^T\mathcal{V}_4 = 0 \ (17), \\
w^T\mathcal{C}^5 = 1 \ (6), & w^T\mathcal{A}^{-1}(\mathcal{C}^2.U_4) = 0 \ (18), \\
w^T\mathcal{T}_1 = 0 \ (7), & w^T\mathcal{A}U_4 = 0 \ (19), \\
w^TU_4 = 0 \ (8), & w^T\mathcal{V}_1 = 0 \ (20), \\
w^T\mathcal{T}_2 = 0 \ (9), & w^T\mathcal{V}_3 = 0 \ (21), \\
w^T\mathcal{A}^{-1}(\mathcal{C}.U_4) = 0 \ (10), & w^T\mathcal{C}.\mathcal{A}^{-1}(\mathcal{C}.U_4) = 0 \ (22), \\
w^T\mathcal{C}^6 = 0 \ (11), & w^T\mathcal{V}_5 = 0 \ (23).
\end{array}
$$

Proposition 4

1. *$\mathcal{V}_1$, $\mathcal{V}_2$, $\mathcal{A}U_4$ and $\mathcal{V}_3$ are linearly dependent.*
2. *$\mathcal{V}_1$, $\mathcal{V}_2$, $\mathcal{A}U_4$ and $\mathcal{V}_4$ are linearly dependent.*
3. *$\mathcal{V}_1$, $\mathcal{V}_2$, $\mathcal{A}U_4$ and $\mathcal{V}_5$ are linearly dependent.*
4. *$\mathcal{V}_1$, $\mathcal{V}_2$, $\mathcal{A}U_4$ and $\mathcal{V}_6$ are linearly dependent.*
5. *$\mathcal{V}_1$, $\mathcal{V}_2$, $\mathcal{A}U_4$ and $\mathcal{V}_7$ are linearly dependent.*

Proof: Because of the expression of the vectors $\mathcal{V}_1$, $\mathcal{V}_2$, $\mathcal{A}U_4$ and $\mathcal{V}_3$, it is sufficient to show that v_1, v_2, Au_4 and v_3 are linearly dependent (idem for the part 2 to 5 of the proposition). The method $R = (A, b, c)$ is of local order 8 for the differential component. Thus, order conditions associated with the trees of $DAT2_y(7)$ are satisfied. In particular,

$$
\begin{array}{llll}
b^TA^{-1}c^7 = 1, & b^TA^{-1}(c.A^2c^4) = \frac{1}{30}, \\
b^TA^{-1}(c.Ac^5) = \frac{1}{6}, & b^TA^{-1}(c.A(c.A^{-1}c^5)) = \frac{5}{6}, \\
b^TA^{-1}(c^2.Ac^4) = \frac{1}{5}, & b^TAc^5 = \frac{1}{42}, \\
b^TA(c.A^{-1}c^5) = \frac{5}{42}, & b^Tc^6 = \frac{1}{7}, \\
b^Tc.Ac^4 = \frac{1}{35}, & b^Tc.A^{-1}c^6 = \frac{6}{7}, \\
b^Tc.A^{-1}(c.Ac^4) = \frac{6}{35}, & b^Tc^2.A^{-1}c^5 = \frac{5}{7}.
\end{array}
$$

Finally, we obtain

$$
\begin{array}{llllllllll}
b^Tv_1 & = & b^Tv_2 & = & b^Tv_3 & = & b^Tv_4 & = & 0, \\
b^Tv_5 & = & b^Tv_6 & = & b^Tv_7 & = & b^TAu_4 & = & 0,
\end{array}
$$

but $b \neq 0$, hence the proposition is shown. □

Remark 4 *If the step size is constant (i.e. $r_1 = \cdots = r_p$), then we have*

1. *$\mathcal{V}_1$, $\mathcal{V}_2$, $\mathcal{A}^{-1}U_4$ and $\mathcal{V}_4$ are linearly dependent.*
2. *$\mathcal{V}_1$, $\mathcal{V}_2$, U_4 and $\mathcal{V}_5$ are linearly dependent.*
3. *$\mathcal{V}_1$, $\mathcal{V}_2$, $\mathcal{A}U_4$ and $\mathcal{T}_1$ are linearly dependent.*
4. *$\mathcal{V}_1$, $\mathcal{V}_2$, $\mathcal{A}U_4$ and $\mathcal{T}_2$ are linearly dependent.*

Hence, nine equations are identically satisfied and p can be choosen equal to four.

References

[BC93] K. Burrage and R.P.K. Chan. *On smoothing and order reduction effects for implicit Runge-Kutta formulae.* Journal of computational and Applied Mathematics 45, 1993. North-Holland, CAM 1275.

[BCP91] K.E. Brenan, S.L. Campbell, and L.R. Petzold. *Numerical solution of initial value problems in differential-algebraic equations.* North Holland, 1991. New York.

[CC95] R. Chan and P. Chartier. *A Composition Law for Runge-Kutta Methods Applied to Index-2 Differential-Algebraic Equations.* IRISA, Fevrier 1995. Publication interne n.893.

[Hig93] I. Higueras. *Coefficients of the Taylor expansion for the solution of differential-algebraic systems.* North Holland, 1993. Applied Numerical Mathematics 12, page 497-501.

[HLR89] E. Hairer, C. Lubich, and M. Roche. *The Numerical Solution of Differential Algebraic Systems by Runge-Kutta Methods.* Springer-Verlag, 1989. Lecture Notes in Mathematics 1409.

[HNW91] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations (Vol 1).* Springer-Verlag, 1991. 2nd Edition.

[HW91] E. Hairer and G. Wanner. *Stiff Problems and Differential Algebraic Problems (vol.2).* Springer-Verlag, 1991. 2nd Edition.

[Pet86] L.R. Petzold. *A description of DASSL: A differential/Algebraic System Solver.* Proceedings of IMACS World progress, 1986. Montreal, Canada.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399